

# Automated Teleprompter

M.M.S.A. Paris, G.W.G.K.N. Udayanga, W.M.T.C.B. Weerakoon, K.W.C. Mihiran, C.R.D. Silva  
Department of Computer Science and Engineering,  
University of Moratuwa,  
Moratuwa, Sri Lanka.

**Abstract-** This project is focused on automating the current manual teleprompting system. Teleprompting is one of the most widely used techniques in many broadcasting systems. Nowadays, the teleprompting process is handled manually. Our purpose is to automate the teleprompting scenario, so that no manual intervention is needed to operate the process. This project mainly consists of two sections; Teleprompter Automation and Graphical implementation for sign language to illustrate the speeches or conversion for the handicaps.

**KeyWords** – Teleprompter, Sign language, Signal processing, MATLAB, C#.

## I. INTRODUCTION

In public speaking it is very important to appear confident about the speech that you are going to deliver. People give more attention to speakers who show confidence in the field they are talking about. So it is necessary to keep your main point intact. However, when the duration of the speech increases, the amount of detail also increases making it hard to memorize the speech. Also it does not look a spontaneous speech due to presenter trying to remember the points of the speech which not give a good impression to the audience. Using a Teleprompter enables the speaker to be confident about the speech he is going to deliver even if he cannot remember the exact details of the speech. He can also maintain eye contact with the audience while looking at the script.

The main reason for the emerging of the project's idea of Teleprompter Automation is to make the teleprompting process much easier for the people in the broadcasting world. In the present world, the teleprompting is not an easy task for a person or set of people to control manually in the real time as there might be situations where the process is done with severe mistakes by the operators as well as the hardness of the controlling. But if the process is automated the process becomes less prone to errors and also much easier for the administrators to handle with higher efficiency.

The other section of Graphical implementation for sign language to illustrate the speeches or conversions for the handicaps is mainly due to the intention of facilitating the handicapped people. The proposed graphical implementation is supposed to be very efficient for many kinds of audiences who really need a proper support from the current technology.

## II. LITREATURE REVIEW

### A. Teleprompter

A teleprompter is a display device that prompts the person speaking with an electronic visual text of a speech or script. The screen is in front of the performer, and the words on the screen are reflected to the eyes of the performer, normally using a sheet of clear glass.

Teleprompter system mainly consists of a video monitor displaying the script and a clear glass to reflect it to presenter's eyes without refracting it to the audience. This is the usual setup for Teleprompters used in public speeches. But in the field of news reporting it is necessary for the news presenter to keep direct eye contact with the video camera. So a hardware setup as shown in Fig. 1 needs to be implemented to execute the teleprompting process without losing the eye contact with the camera.

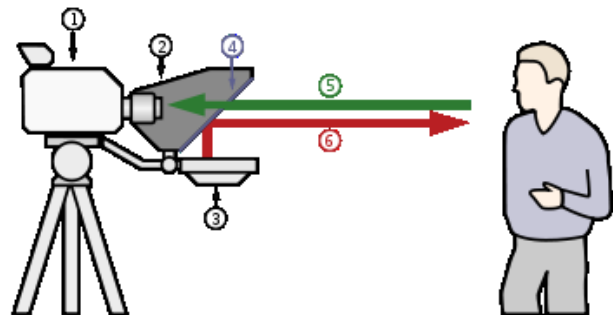


Figure 1: Schematic representation of a Teleprompter[1]

- (1) Video camera
- (2) Shroud
- (3) Video monitor
- (4) Clear glass or beam splitter
- (5) Image from subject
- (6) Image from video monitor

### B. Signal processing

Signal processing involves the filtering of human voice range frequencies from the audio input of the presenter's microphone. As the input from the microphone contains frequencies out of human voice range, it is necessary to filter them before analyzing the voice activity. This step ensures that the accuracy of the calculations ahead becomes very high as the noise frequencies do not have any effect on the final result.

MATLAB facilitates many built-in audio processing functions which are required for the project. Real time wave recording and Butterworth filtering can be given as

examples for them. In addition to that, the capability of graphical representation of large data sets in subplot graphs in MATLAB was helpful.

Wavrecord function is an in-built MATLAB function for recording sound using PC based audio input device [2]. It is useful in live capturing presenter’s voice and storing it for sound processing. Wavrecord function captures the sound and stores it in matrix form in a preferred data type.

A Band-Pass filter is a device that accepts frequencies within a certain range and rejects (attenuates) frequencies outside that range [3]. This is a combination of a Low-Pass filter and a High-Pass filter to extract only the human voice range of 200Hz – 7000Hz.

“Butter designs lowpass, bandpass, highpass, and bandstop digital and analog Butterworth filters. And Butterworth filters are characterized by a magnitude response that is maximally flat in the passband and monotonic overall” [4]. Butter function [5][6] can be used in highpass and lowpass filtering modes which is essential for removing non vocal sounds from the sound input.

Voice activity detection (VAD), also known as speech activity detection or speech detection, is a technique used in speech processing in which the presence or absence of human speech is detected [7].

### C. Sign Language related study

A sign language is a language which, instead of acoustically conveyed sound patterns, uses visually transmitted sign patterns (manual communication, body language) to convey meaning—simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to fluidly express a speaker’s thoughts [8][9].

Sign languages are on different standards as some of them represent a word with a sign and others represent a letter in a sign. For a real-time system like ours it is efficient to use the sign per word representation. In most cases, there is a one to one map between words and signs. In some occasions there are signs which represent number of words.

## III. DESIGN AND IMPLEMENTATION

The basic research for the automated teleprompter project was done using matlab. With the aid of matlab we researched about voice to text mapping which is the core requirement in scrolling the teleprompter screen. The other side of the research was carried out for the avatar implementation. For that we have to graphically illustrate the words in sinhala language, since the project was basically focused on that language. Finally we want to convert all the MatLab code to C#. Through the C# we linked the voice to text mapping part and the text to sign language video library. Fig. 2 shows the basic functionalities carried out in this project. Following subsections describes the methodologies that we involved in our project.

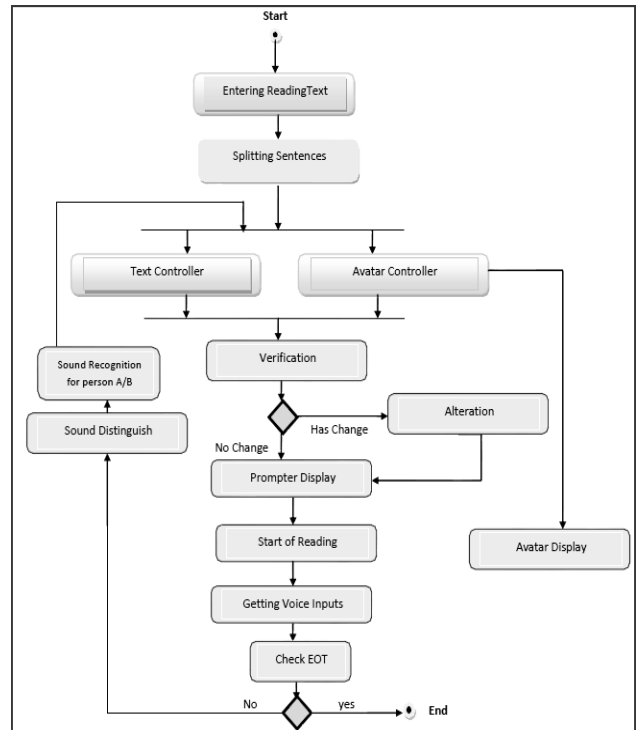


Figure2: A high level representation for Basic functionalities for automated teleprompter

### A. MATLAB prototype

To understand the primitive requirements of accurate voice recognition, for word detection we decided to implement a prototype based on “real time word detection”. Since this prototype development had a considerable research area, we used MATLAB language which is optimized for engineering research based simulations rather than usual development base programming languages. Basically code for the prototype was encapsulated within two m-files called Simulate.m and Splitter.m.

Simulate m-file contains the functionality for voice recording. Since 8000Hz discrete voice capturing (8000 samples per second) is sufficient to recognize the continuous human speech, we did the recording in ‘mono’ mod at this frequency. Then the recorded voice data for each one second is band pass filtered so as to preserve the 200-7000Hz frequency range which collects the human voice frequencies. As the band pass filter we use the Butterworth filter of MATLAB. The filtered data is collected by the one dimension matrix. When obtaining discrete voice samples there are some zero amplitude voice samples between the letters of the word itself. So we need to consider something like correlation among voice samples since it rebuilds the continuous nature of the human speech. And also many differences/variations in human voice can be captured with voice energy, by obtaining the energy of each sample collectively with samples around the required samples [10][11]. After obtaining the energy variation of samples we can apply our magnitude threshold to obtain a step function showing a binary output for speaking and pausing times. We input

this to splitter function which contains the logic for determining the existence of a word.

Splitter m-file is basically a function which accepts the stepFn and many other variables from the simulate function and pump those to logical pipes that we use for word detection. The return values of this function contain the desired output (the wordFn) and many other synchronization and chaining variables among the 1 second processing periods.

By nature, when a human is speaking there are silent gaps/pausing intervals between letters, words as well as between sentences in increasing order. We can detect a start of a word with appropriate threshold values for pausing time between words. But there are some silent intervals which are same size or larger than the silent threshold within a speaking word itself because of the syllables and so on. So we need to consider a rough amount of time that has to be spent to speak a particular word. If we can have a rough value for a minimum time (in samples) to speak a letter and if the word length is given we can calculate a rough time that will spend to speak that word at the fastest possible speaking. Obtaining a value for the minimum threshold for the time spent to speak a letter is very important. Since we have the script we can track the expected word and its length. In our logic, if we detect a starting of a word we assign the silent length it has before speaking, to a temporary variable. Then we check whether the speaker speaks more than 100 samples. In detecting a word we check whether the speaktime is greater than the expected word length. If it is greater than the expected word length a word is detected. Fig. 3 shows the implementation of Matlab prototype.

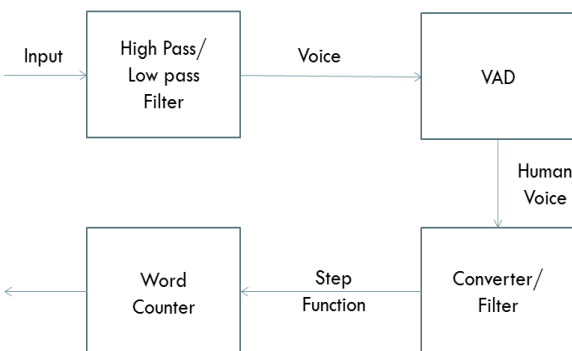


Figure3: the implementation of Matlab prototype

### B. Live voice capturing and real time voice processing

For real time audio recording purpose, we use the methods of the “NAudio” audio library which is for C#. For each second we take 8000 samples of live voice. We send the voice samples for processing at each 1s to the filter.

### C. Separation of human audible frequency range and noise filtering using sound processing module

To separate the human voice we use the Butterworth filter [4]. With that we obtained the frequency components between 200-7000Hz which cover the frequencies of human voice. To eliminate the noise within this frequency range we calculate the energy [10][11] of each sample and use a threshold for the magnitudes of the spectrum. And to make the analyzing process easier, we develop a step function which makes the magnitude of voice components to zero while real speaking sound components to one. To compensate for both of these values correctly we use a magnitude threshold.

### D. words counting to script scrolling module

After using the magnitude threshold we get a function with voice active regions and inactive regions for each second. Then we execute wordSplit function which contains the algorithm developed in MATLAB prototype. If the voice inactive region exceeds a threshold value, (obtained by research) the next voice active region is considered as a word and the word counter is incremented. These three steps are shown in Fig. 4.

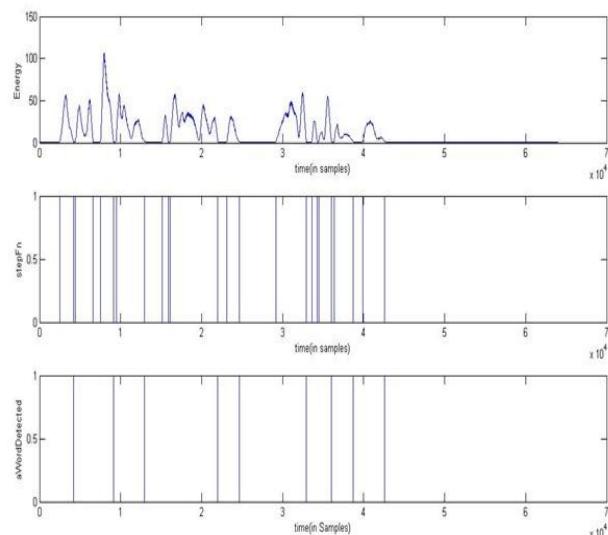


Figure4: 1. Energy variation of speech signal 2. Step function 3. Word positions

Script scrolling Module handles importing of the Script as a .TXT file to the program. The .TXT file should be in Unicode encoding, as the language which the program developed is for Sinhala. So it only supports a Script which is encoded in Unicode and in Sinhala Unicode range. First Script will be passed through filters to remove unnecessary characters from the Script, and to format the Script accordingly. Then the Script should be split to

Sentences and Words for the ease of displaying the flow of words in the Teleprompter Interface. This task is also carried out by the Script Controlling Module. Here the Sentences and Words from the Script is split accordingly and are indexed into array to allow easy access of a Word from a certain Sentence.

#### E. Sign language avatar and Avatar controller module

Sign languages are on different standards as some of them represent a word with a sign and others represent a letter in a sign. For a real-time system like ours it is efficient to use the sign per word representation. In most cases there is a one to one map between words and signs. In some occasions there are signs which represent number of words [8]. Sign language movements were simulated using a sign language video library.

The Avatar controller module is mainly controlled according to the Script controller module of the system. The Avatar controller module behaves according to the Script controller module in a way such that the position of avatar animation is derived from the script position of the given text.

Main functionality of avatar controller module is to get the current script position from the Script Controller Module and then check whether that detected Sinhala Unicode word exists in the property file and call it from the mapped position of the video database if it exists. Then send that video to the Avatar Window to play in order to show the sign language posture to the viewer.

### IV. RESULTS

There are many software [12][13][14] available for the teleprompting scenario in the literature. But all of them use a manual timer and a separate human controller to scroll the script according to the presenter's voice. Here main disadvantage is that it can run faster or slower than the presenters speed. Our system was intended to automate this scrolling and reduce the out of sync situation. It is the only system that uses presenter's voice input to track the position of the script.

Using presenter's voice provides automation and many other possibilities. With the script in hand we tried to get as much as information from the script to do our automation. This also allowed us to implement the Sign language avatar synchronized with the presenter's voice. Also real time prompts messages for director to communicate with presenter. Fig. 5 shows the interface of Automated Teleprompter.

The Matlab prototyping of the system brings some promising results with some restrictions. It performs the word splitting process accurately within some constrains. The accuracy of the system is dependent on the speaker's voice characteristics. Tampering with constants in the prototype, the system could achieve higher accuracy in word splitting process. First parameter we have to estimate is the time of letters which specifies the

minimum time needed to speak a single letter. From our test simulation we found this value to be approximately 250 samples with respect to our sampling rate. Also the silent threshold to split the words was calculated to be 430 samples with respect to our sampling rate. These values were calculated for different speakers with different speed to get a good statistical average.

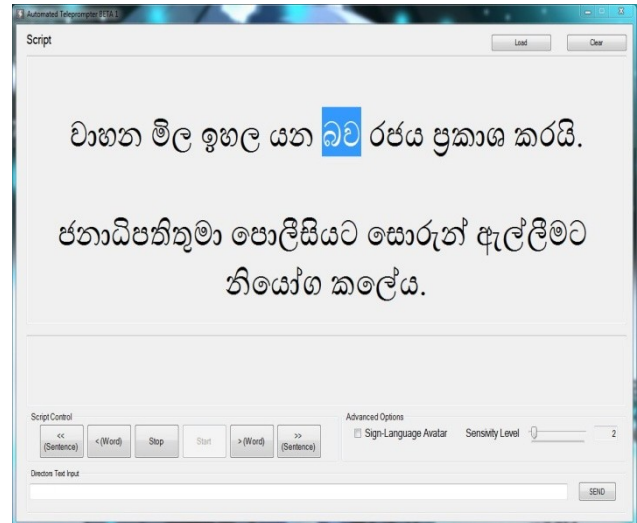


Figure 5: GUI module of Automated Teleprompter

Changing the word detection magnitude threshold and voice inactive interval (silent threshold) are the main two parameters that give some impressive results. When speaker is speaking somewhat slow rate, the word detection process get executed very successfully and captured all of the words. In case a word is not properly detected there seem to be a lag of 2-3 words and controls are provided to maintain the synchronization.

Sign language avatar (Fig. 6) is a very innovative idea. Currently there is no sign language video library available for Sinhala language. Most of the time it is done manually and it has a considerable amount of lagging behind the voice. Our attempt was to create a Sinhala sign language video library using available video clips and use that library to present the speech in sign language. This is implemented using a property file for more extensibility in words or convert into other languages. The result of the sign language avatar was a success due to the iterative development we undertook and frame rate.



## V. CONCLUSION

Automated Teleprompter surpassed our expectations and achieved lot more achievements in multiple fields. Although the accuracy of the system is not 100%, we were able to achieve very high standards for a semi-automated one. One thing we learnt from our research was that it is very hard to achieve full automation without performing a training session and analyzing presenter's voice which exerts a huge overhead for speech recognition. So we can safely say that an optimal output was maintained by our method by allowing presenters to use the system without any initial training.

## VI. FUTURE WORK

Although we have provided a whole lot of functionality as we have declared before, there are still more that could be done to further develop the Automated Teleprompter project. So far we have covered almost all the basic requirements needed to automate the teleprompting and also the avatar animation process, mapped to the given script. But if we think a bit out of the scope and try to make it even better by adding some other advanced requirements, it would be really good for its improvement in the future, although it is a complete product to be used at this stage.

There are quite a few areas that we can extend our scope and the following areas represent some such scenarios.

At this stage the system only supports for the Sinhala Unicode. The system is developed to support for the Sinhala scripts to be read by speakers. The system can be further developed to support for other language as well using relevant Unicode systems.

Signal processing is one of the most important areas in this project since the automation of the teleprompter thoroughly depends on how the input audio signal is processed. The existing accuracy rate of the signal processing mechanism is approximately 85% after going through a lot of filtering and other mechanisms. But anyway if this process can be further developed to increase the accuracy rate it will vastly enhance automation quality of the project.

Avatar animation of the sign language for the disable people is also another very important area in this project. The avatar controller unit depends on the script controller for it to get the current positions of the script to enable real time position videos. But the checking, calling and processing for videos are totally handled by the avatar controller unit. If the existing implementation can be further fine-tuned to optimize the checking, calling and processing scenarios then it can upgrade the performance of the system and memory management.

Authors of this paper would like to thank Dr. Shantha Fernando for coordinating this module for us. Special thanks go to all the staff and the support staff members of Department of Computer Science and Engineering, University of Moratuwa and all our fellow batch mates for their support.

## REFERENCES

- [1] Teleprompter [Online] [Accessed: May 06, 2011] <http://en.wikipedia.org/wiki/Teleprompter>
- [2] wavrecord [Online] [Accessed: May 20, 2011] <http://www.mathworks.com/help/techdoc/ref/wavrecord.html>
- [3] Band-Pass filter [Online] [accessed: May 06, 2011] [http://www.electronics-tutorials.ws/filter/filter\\_4.html](http://www.electronics-tutorials.ws/filter/filter_4.html)
- [4] Butterworth filter design [Online] [Accessed: May 20, 2011] <http://www.mathworks.com/help/toolbox/signal/butter.html>
- [5] Wireless Engineer (also called Experimental Wireless and the Wireless Engineer), vol. 7, 1930, pp. 536–541
- [6] Butterworth, S. On the Theory of Filter Amplifiers [www2.ee.ufpe.br/codec/paper%20BUTTERWORTH.pdf](http://www2.ee.ufpe.br/codec/paper%20BUTTERWORTH.pdf)
- [7] Ramirez, J.J. M. Górriz, J. C. Segura. Voice Activity Detection. Fundamentals and Speech Recognition System Robustness. 2007.
- [8] Sign language [Online] [Accessed: May 21, 2011] [http://en.wikipedia.org/wiki/Sign\\_language](http://en.wikipedia.org/wiki/Sign_language)
- [9] Rohana Special School [Online] [Accessed: May 18, 2011] <http://www.rohanaspecialschool.org/>
- [10] Energy (signal processing) [Online] [Accessed: July 03, 2011] [http://en.wikipedia.org/wiki/Energy\\_%28signal\\_processing%29](http://en.wikipedia.org/wiki/Energy_%28signal_processing%29)
- [11] Michael Peter Norton and Denis G. Karczub. Fundamentals of Noise and Vibration Analysis for Engineers. 2003. Cambridge University Press.
- [12] Free Teleprompter [Online] [Accessed: Oct 03, 2010] <http://www.freetelepromptersoftware.com/>
- [13] PropmtDog [Online] [Accessed: Oct 03, 2010] <http://www.promptdog.com/>
- [14] SignGenius [Online] [Accessed: Oct 03, 2010] <http://www.signgenius.com/signlanguage-american.shtml>